

# Penerapan Rumus Haversine Dalam Pencarian Produk Terdekat Berbasis *Google Map API*

**Pratikto Aditia W**

Sistem Informasi, Fakultas Ilmu Komputer  
Universitas Sriwijaya Palembang  
[Pratikto63@gmail.com](mailto:Pratikto63@gmail.com)

**Ali Ibrahim**

Sistem Informasi, Fakultas Ilmu Komputer  
Universitas Sriwijaya Palembang  
[aliibrahim.ilkom.unsri.ac.id](mailto:aliibrahim.ilkom.unsri.ac.id)

**Abstrak**--Semakin pesatnya ilmu pengetahuan dan teknologi khususnya internet perlahan demi pasti akan merubah gaya hidup seseorang untuk membeli produk dari sistem pembelian konvensional ke sistem pembelian *online*. Namun kadangkala tidak semua konsumen percaya akan produk yang ditawarkan secara *online* hanya dengan menampilkan data-data produk secara digital seperti foto, *video* dan lain-lain. Perlu peninjauan langsung dari pembeli ke penjual untuk melihat produk secara fisik. Penelitian ini menggunakan rumus Haversine untuk menentukan jarak terdekat, kemudian teknologi yang digunakan adalah teknologi *geolocation* yang merupakan sebuah *API* dari *HTML5* menggunakan *Google Map* yang memungkinkan untuk mengetahui lokasi penjual berdasarkan koordinat latitude dan longitude keberadaannya. Penelitian ini akan menyajikan informasi jarak terdekat dari pembeli kepenjual untuk peninjauan langsung bentuk fisik produk tersebut dan merekomendasikan produk yang akan dibeli sesuai jarak terdekat, dengan memanfaatkan *geolocation* dan *haversine formula* dalam sebuah sistem informasi geografis.

**Kata kunci** : *Geolocation*, *Haversine Formula*, *GoogleMap*

## I. Pendahuluan

Semakin berkembangnya ilmu pengetahuan dan teknologi informasi terutama internet membuat sebagian besar orang terutama pelaku usaha kecil dan menengah berlomba-lomba untuk menjual dan memasarkan produknya secara mudah dan cepat dengan pangsa pasar yang luas dan memperkecil biaya operasional dan waktu jual yang tidak terbatas. Kenyataan tersebut membuat proses digitalisasi penjualan berbasis *website* semakin meningkat. Permasalahan yang terjadi tidak semua pelaku usaha

diperuntukan untuk Layanan Berbasis Lokasi (LBS), dengan adanya teknologi ini pengguna sistem dimungkinkan untuk mengetahui lokasinya berdasarkan latitude dan longitude keberadaannya. Berbagai macam metode dan persamaan atau formula yang digunakan untuk mengetahui jarak antara dua titik, *Haversine Formula* merupakan salah satu persamaan yang sangat akurat untuk menentukan jarak antara dua titik di bumi. Pemanfaatan *Geolocation* dan *Haversine Formula* dalam sebuah SIG bertujuan agar pengguna sistem dapat melakukan perhitungan seberapa dekat jarak pembeli dan penjual. Penelitian ini akan menyajikan informasi jarak terdekat dari pembeli kepenjual untuk peninjauan langsung bentuk fisik produk tersebut. Dan merekomendasikan produk yang akan dibeli sesuai jarak terdekat. Dari permasalahan tersebut penulis melakukan penelitian ini bertujuan untuk memberikan informasi geografis pencarian produk terdekat dari konsumen menggunakan menggunakan rumus *Haversin* berbasis *GoogleMap API*.

## II. Studi litelatur

Banyak sumber acuan yang digunakan untuk membangun sistem informasi geografis ini. Baik dari buku-buku ataupun dari penelitian-penelitian sebelumnya. Penelitian yang dilakukan oleh Gintoro (2010) yang berjudul "Analisis dan Perancangan Sistem Pencarian Taksi Terdekat dengan Pelanggan menggunakan Layanan Berbasis Lokasi" di rancanglah sebuah aplikasi yang memberikan lokasi-lokasi pelanggan pemesanan taksi. Sistem tersebut menangkap lokasi pemesanan taksi yang menggunakan *smartphone BlackBerry* dengan menggunakan *GPS* didalamnya dan dengan cara menghitung lokasi koordinat taksi dan pelanggan menggunakan *Haversine Formula*[2].

### a. Rumus Haversine

Rumus haversine adalah persamaan yang penting pada navigasi, memberikan jarak lingkaran besar antara dua titik pada permukaan bola (Bumi) berdasarkan bujur dan lintang. Berikut adalah rumus Haversine [1].

produknya di *website-website* iklan baris secara gratis. Namun yang terjadi saat ini kadangkala konsumen tidak sepenuhnya percaya dengan data-data digital baik foto maupun *video* produk yang dipasarkan didalam *website* iklan baris. Perlu peninjauan langsung dari pembeli ke penjual untuk melihat produk secara fisik. Untuk meningkatkan kepercayaan pembeli terhadap penjual maka dibentuklah sistem informasi geografis lokasi penjual. Salah satu teknologi yang berkaitan erat dengan sistem informasi geografis adalah teknologi *Geolocation* yang merupakan sebuah *API* dari *HTML5* yang

$R$  = Radius bumi (rata-rata radius = 6,371 kilometer)

$$\Delta \text{lat} = \text{lat2} - \text{lat1}$$

$$\Delta \text{long} = \text{long2} - \text{long1}$$

$$a = \sin^2(\Delta \text{lat}/2) + \cos(\text{lat1}) \cdot \cos(\text{lat2}) \cdot \sin^2(\Delta \text{long}/2)$$

$$c = 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{1-a})$$

$$d = R \cdot c$$

Sudut pada rumus menggunakan satuan radian untuk menggunakan fungsi trigonometri. Rumus *harvesine* digunakan dalam penelitian ini untuk melakukan perhitungan jangkauan jarak terdekat produk penjual terhadap jarak pembeli.

#### b. Sistem Informasi Geografis

Sistem Informasi Geografis merupakan aplikasi yang memiliki banyak kegunaan. Banyak aktivitas manusia dalam berbagai bidang yang akan sangat terbantu apabila aplikasi *GIS* diimplementasikan dengan baik. Sistem informasi geografis saat ini menjadi sistem yang penting dan menarik dan sedang banyak dipelajari di berbagai bidang kebutuhan kerja. Sistem Informasi Geografis memiliki kemampuan dasar sebagai *mapping system* dengan kemampuan kartografinya dalam menjawab hal-hal terkait analisis (*query*) [5].

#### c. Google Maps API

*GoogleMaps API* menyediakan sejumlah utilitas untuk memanipulasi peta dan menambahkan konten didalam peta melalui berbagai layanan, memungkinkan pembuatan aplikasi map yang kuat pada situs web [3]. *GoogleMaps API* merupakan perkembangan dari *google Maps*. Dengan menggunakan *google Maps API* ini, dimungkinkan untuk dapat menggunakan *google Maps* di dalam website. Meski awalnya hanya *JavaScript API*, *Maps API* diperluas untuk menyertakan sebuah API untuk aplikasi Adobe Flash. Keberhasilan *GoogleMaps API* telah melahirkan sejumlah pesaing antara lain *Yahoo! Maps API*, *Bing Maps Platform*, *MapQuest Development Platform* dan *OpenLayers* [6].

### III. Metodologi Penelitian

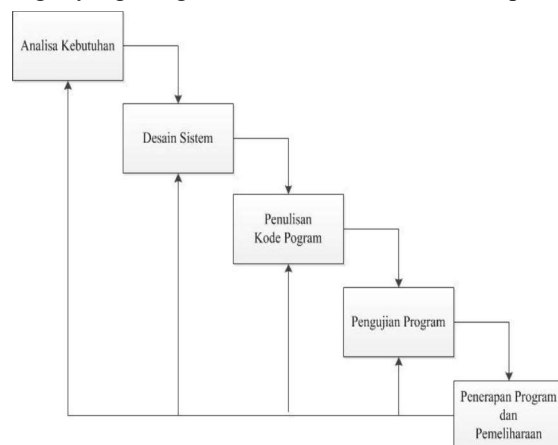
Alur perancangan aplikasi ini dengan metodologi waterfall. Metode waterfall dipilih karena sesuai dengan kebutuhan sistem yang akan dibangun. Fase dari metode waterfall dapat dilihat pada gambar 1 berikut:

Adapun fase-fase dalam metode ini adalah sebagai berikut:

#### A. Analisa Kebutuhan

Langkah ini merupakan analisa terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini bisa melakukan sebuah penelitian, wawancara atau studi literatur. Sistem analis akan menggali informasi sebanyak-banyaknya dari user sehingga akan tercipta

sebuah sistem komputer yang bisa melakukan tugas-tugas yang diinginkan oleh user tersebut. Tahapan ini



Gambar 1. *Waterfall Model* (Pressman, R. 2010)

akan menghasilkan dokumen user requirement atau bisa dikatakan sebagai data yang berhubungan dengan keinginan user dalam pembuatan sistem.

#### B. Desain sistem

Tahap selanjutnya adalah perancangan sistem yang akan menggambarkan fungsional dari sistem yang akan dibangun

secara keseluruhan. SIG pencarian produk terdekat dengan konsumen yang dibangun berbasis web dan menggunakan peta dasar Google Maps. Fungsional yang dibangun akan disesuaikan dengan analisa kebutuhan yang telah dilakukan sebelumnya.

#### C. Penulisan Kode Program

Penulisan kode program atau coding merupakan penerjemahan design dalam bahasa yang bisa dikenali oleh komputer. Penulisan program bisa dimulai dari menulis program GoogleMap API dengan urutan sebagai berikut (1). Memasukkan *Maps API JavaScript* ke dalam *HTML*, (2). Membuat element div dengan nama *map\_canvas* untuk menampilkan peta, (3). Membuat beberapa objek literal untuk menyimpan properti-properti pada peta, (4). Menuliskan fungsi *JavaScript* untuk membuat objek peta, (5). Meng-inisiasi peta dalam *tag body HTML* dengan event *onload* [4].

#### D. Pengujian Sistem

Pada tahap ini program yang telah dibuat dan diuji per unitnya kemudian disatukan menjadi suatu sistem yang utuh dan diuji secara keseluruhan guna menguji tingkat integrasi antar unit yang dibuat sebelumnya.

#### E. Maintenance atau Perawatan





Pada tahap ini penulis tidak menerapkan tahapan ini karena perangkat lunak baru saja

tampung pada *variabel* program php yang akan dijadikan perbandingan dalam rumus *Haversine* untuk pencarian produk terdekat dari konsumen. Percobaan dan hasil dari implementasi rumus *haversine* untuk pencarian produk terdekat dari konsumen dapat dilihat pada tabel 1 di bawah ini.

Tabel 1. Hasil Percobaan

No	produk	Penjual	Jarak (Km)
1	sepeda	kakkooi	1 Km
2	komputer HP	amir04	2 Km
3	LAPTOP ACER TERBARU	amir04	2 Km
4	hp samsung s4	amir04	2,3 Km
5	memori terbaru	budi03	8 Km
6	rumah termurah	budi03	9 Km

Options									
			id	username	user_type	password_hash	target_email	email	status
			1	pratik01	user	\$2\$12\$8v14X0F67G7q1T6063H8nPa3pG3uM5kd0C3C8r	kateon03@gmail.com	0	
			2	kateon03	user	\$2\$12\$8d32ymACa2u65597vdyQyU4W57uW2u6P6K6	pratik01@gmail.com	0	
			3	tush03	user	\$2\$12\$7d9514Pw38n2G47P3M1R6p0uaww15y1	tush@gmail.com	0	
			4	amr04	user	\$2\$12\$87m0u0L13488L1J5w2u61034482d3u0u04	amr@gmail.com	0	
			5	anil	user	\$2\$12\$8C8uCh4R4938m1k4u4e053u61P4P4557mD74	anil@gmail.com	0	
			6	tush04	user	\$2\$12\$8u17H0dL1P41e0m47327q67338uG38C08e	tush@gmail.com	0	

Options		id produk	harga	alamat	latitude	longitude
  	2	motor murah	20000000	Judupur, Rajasthan, India	26.2310717025072277	73.193424246535
  	2	sepeda	20000000	Kota Palembang, Sumatera Selatan, Indonesia	-2.991027335742498	104.755046303515193
  	3	motor memori kualitas ok	130000	Jl. Gubernur H. Anwar Mangku Harto, Kebun Bunga, S...	-2.903277447775405	104.726789889843
  	4	rumah ternamur	100000000	Jl. Gubernur H. Anwar Mangku Harto, Kebun Bunga, S...	-2.9006995	104.71821989999999
  	5	hp samsung s4	320000	Jl. Mayor Ruslan No.1190, Oluw, Ir Tim. II, Kota...	-2.963091912877661	104.760079222962579
  	6	komputer HP	345000000	Jl. Mayor Ruslan No.1190, Oluw, Ir Tim. II, Kota...	-2.9639937	104.76009890000002
  	7	LAPTOP Acer TERBARU	200000000	Jl. Mayor Ruslan No.1190, Oluw, Ir Tim. II, Kota...	-2.9639937	104.76009890000002

Dari tabel diatas percobaan dilakukan dengan koordinat *latitude* (-2.9777877972440656) dan *longitude* (104.75714876335451) konsumen pada radius 10 km. Didapat hasil 6 produk yang dijual berada disekitar lokasi konsumen. Untuk menampilkan data tersebut, ada beberapa file yang dibutuhkan yaitu file koneksi ke database mysql, file index, dan file tampil. File koneksi memanggil database quickad yang berada diserver localhost dengan username root dan tanpa passowrd. Dengan file koneksi tersebut semua data yang tersimpan di tabel penyimpanan produk bisa ditampilkan. Selanjutnya file index adalah file yang berfungsi untuk menampilkan peta google map pada browser. File index mutlak di gunakan untuk menjadikan file default pada website. Kemudian file tampil adalah file yang berfungsi untuk menampilkan berkas xml dengan dengan memanfaatkan echo php. Dengan adanya file tampil tersebut semua produk yang lokasi jarak terdekat dari konsumen dapat ditampilkan.

Dari hasil yang diperoleh dapat ditarik kesimpulan sebagai berikut :

1. Sistem informasi *geografis* menggunakan *GoogleMap* ini dapat menentukan lokasi *geografis* penjual.
2. Pencarian produk terdekat dari konsumen dengan perbandingan koordinat *latitude* dan *longitude* konsumen menggunakan rumus *haversine* yang telah diimplementasikan dalam kode program.
3. Dengan sistem informasi *geografis* ini dapat juga merekomendasikan produk yang akan dibeli konsumen berdasarkan jarak terdekat dari area konsumen.

```

to complete: [Tab]->Next Tag. [Ctrl+Space]->List All Tags. [Ctrl+Enter]->List All Tags.

SELECT ad_product.id AS id, ad_product.product_name AS product_name,
       ad_user.username AS penjual, (6371 * ACOS (
       COS(RADIANS('-2.9777877972440656')) * COS(RADIANS(lat)) *
       COS(RADIANS('104.75714876335451') - RADIANS(lng)) +
       SIN(RADIANS('-2.9777877972440656')) * SIN(RADIANS(lat))))
       jarak, location FROM ad_product, ad_user
WHERE ad_product.user_id=ad_user.id
HAVING jarak < 10

ORDER BY jarak, product_name LIMIT 999;

```

## DAFTAR PUSTAKA

- [1]. *Calculate distance and bearing between two Latitude/Longitude points using.* (2017). Dipetik oktober 04,

- 2017, dari movable type script: <http://www.movable-type.co.uk/scripts/latlong.html>
- [2]. Gintoro. (2010). *analisi dan perancangan sistem pencarian lokasi taksi terdekat dengan pelanggan menggunakan layanan berbasis lokasi*.
- [3]. Google. (2017). *Google Map API-Google Code*. Dipetik oktober 04, 2017, dari Google: <https://developers.google.com/maps/?cs=1>
- [4]. Yuhana, U. L. (2010). Pemanfaatan GoogleMaps untuk pemetaan dan pencarian data perguruan tinggi negeri indonesia. *SISFO-Jurnal Sistem Informasi Management Press, Norcross, GA*.
- [5]. Peter A. Burrough, R. A. (2015). Principles of Geographical Information Systems. Dalam R. A. Peter A. Burrough, *Principles of Geographical Information Systems* (hal. 1-23). OUP Oxford.
- [6]. pramatha, I. M. (2012). Implementasi aplikasi sistem informasi geografis dalam pengolahan data jumlah penduduk berbasis web. *Jurnal elektronik Ilmu Komputer - Universitas Udayana, JELIKU V*